*International Journal of Computational and Experimental Science and ENgineering (IJCESEN)*

*Vol. 11-No.2 (2025) pp. 2918-2932*
*http://www.ijcesen.com*

***ISSN: 2149-9144***



**Research Article**

# Chebyshev Polynomial based ElGamal Encryption with Chaotic Greater Cane Algorithm for Secure Communication

**N.V.S.S.Prabhakar[1], Talari Surendra[2]*, G. Narsimlu[3], Subrahmanya S Meduri[4], PSVS Sridhar[5]**

[1]Research Scholar, Department of Mathematics, GSS, GITAM Deemed to be University,Visakhapatnam – 45, Andhra Pradesh, India.
**Email:** prabhakar.nedunuri@gmail.com - **ORCID:** 0009-0000-3042-0688

[2]Assistant Professor, Department of Mathematics, GSS, GITAM Deemed to be University, Visakhapatnam – 45, Andhra Pradesh, India.
* **Corresponding Author Email:** surendrabw.t@gmail.com - **ORCID:** 0000-0002-6511-8152

[3]Assistant Professor, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana
**Email:** gnarsimlu_maths@cbit.ac.in - **ORCID:** 0000-0003-3701-0790

[4]Technical Architect, Wipro Technologies, USA,
**Email:** sumeduri@gmail.com - **ORCID:** 0009-0007-0532-0918

[5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,Vaddeswaram, AP, India,
**Email:** psvssridhar@gmail.com - **ORCID:** 0000-0003-1811-1019

**Article Info:**

**DOI:** 10.22399/ijcesen.1844
**Received :** 05 March 2025
**Accepted :** 16 April 2025

**Keywords**

Chebyshev polynomials
ElGamal encryption
Key exchange
Optimal key selection
Secure communication

**Abstract:**

Recently, the practice of Chebyshev polynomials in public-key system design has been recommended. In fact, they have certain satisfying chaotic features that make them appropriate for usage in cryptography. Thereby, various public-key cryptosystem employing Chebyshev polynomials has been focused however, the successive analysis has revealed its insecurity. In this paper, a novel Chebyshev polynomial based ElGamal Encryption with Diffie- Hellman Key Exchange (CPEE-CFGC) is proposed for guaranteeing security in various applications. The various steps involve in CPEE-CFGC algorithm are key generation, encryption and decryption with secure key exchange process. In the key generation process, the private keys are generated using Fuzzy Logistic Tent Membership Function (FLMF) for each party engaging in the communication. Then, the optimal keys are selected using Greater Cane Rat Algorithm (GCRA). The Diffie Hellman key exchange mechanism is exchange the keys in an unsecure channel. Further, the encryption and decryption process are carried out using chebyshev polynomial based ElGamal encryption (CPEE) algorithm. The simulation of CPEE-CFGC algorithm is carried out using python programming language, and the performance is evaluated with dissimilar performance indicators. As a result, the CPEE-CFGC has obtained a better key generation time of 10256.25 ms, encryption time of 5160.78 ms, decryption time of 230.45 ms and total execution time of 12100.57ms by varying the bit size to 2048 bits than the existing algorithms.

## 1. Introduction

Typically, the design and analysis of cryptography is considered as a mathematical approach that is closely associated with electronic communication and computer technologies in order to ensure secure communication via insecure channels. Despite the great efficacy of conventional symmetric encryption techniques, the confidentiality of ciphertext is lost once the key is exposed, and the security of ciphertext depends only on the key's secrecy in the process of key distribution and administration [1-3]. It is challenging to pass the key between the sending and receiving ends if they are far apart and the key requires to be altered frequently. Diffie and Hellman firstly presented the idea of public-key cryptography to deal the confidentiality issues in symmetric encryption systems, mainly in multi-user communication networks [4, 5].

The first relatively complete public-key cryptography algorithm, RSA was offered by Rivest, Shamir, and Adleman in 1977. Since then, numerous public-key cryptographic algorithms, comprising the ElGamal algorithm, lattice-based cryptography, the elliptic curve cryptography (ECC), the McEliece algorithm, the Mer-Hellman knapsack algorithm, and password-based public-key cryptography have been introduced based on various computational problems [6-8]. One of the most effective and well performing public-key cryptography algorithm still recognized is ElGamal. The difficulty of computing discrete logarithms in a finite field is the base of ElGamal encryption. By permitting the sender to encrypt a plaintext with the public key of receiver and the recipient to decrypt the ciphertext with their private key, it guarantees confidentiality [9-10].

Several ElGamal's encryption techniques have become simple to break owing to the rapid advancement of computer technology. In order to replace or supplement existing public-key cryptographic algorithms, a new practical public-key cryptographic algorithms must be studied since the conventional public-key cryptographic algorithms are continuously encountering different challenges. Meanwhile, the obvious relationship between the fundamental properties of chaos transformations, such as the mixture, sensitivity to parameters and initial values, and cryptography are stated because these properties support well with the major need for secure encryption system. In contrast to block cipher systems and chaotic sequences, the study on chaotic public-key cryptography is still lacking [11], and there are few efficient and practical chaotic encryption techniques. Paying attention to favorable properties of chaotic mappings and classical cryptology principles is still relatively a new field.

A chebyshev polynomial-based public-key encryption technique is stated in [12]. Regretfully, it has determined to be insecure owing to its vulnerability to specific algebraic attacks and inefficient due to its complex calculations. A unique key agreement scheme based on classical RSA method using chaotic mappings and discrete logarithm problems over finite fields [13] is then introduced. This method depends on chebyshev polynomial value over finite fields and ensures key agreement security by evading past active attacks. However, it is challenging to ensure consistency as well as reliable key generation and exchange. Further, an identity-based encryption system with chebyshev polynomials [14] has presented but it exhibits security flaws.

Recently, the chaotic public-key cryptography has become the new advancements [15-18]. In [19], a four-dimensional hyperchaotic map is employed to build a chaos-based cryptosystem with effective substitution and permutation methods. During the substitution stage, the final encrypted image has obtained by effectively XORing the key stream with scrambled images. Nevertheless, when the parameters of chaotic map are not selected correctly, they could have major sensitivity issues and possible weaknesses. A reliable and effective image encryption scheme is suggested in [20] to provide significant security for digital images. It is based on dynamic DNA encoding and DNA operations associated with chaotic maps, such as the henon map, thelorenz system, and the logistic map, with simple structures and highly chaotic behavior. The limitations includes, low randomness, low key space, low sensitivity to plaintext and keys, and lower execution speed. Henceforth, in this paper, an enhanced secure communication based on integrating chebyshev polynomial based elgamal encryption with diffie-hellman key exchange is focused. The major contributions of the proposed CPEE-CFGC algorithm are presented below as follows:

- To propose a novel Chebyshev polynomial based ElGamal Encryption with Diffie-Hellman Key Exchange (CPEE-CFGC) for securing data with higher confidentiality.
- To provide Fuzzy Logistic Tent Membership Function (FLMF) for generating the private keys used for encryption.
- To provide a metaheuristic optimization algorithm, Greater Cane Rat Algorithm (GCRA) for selecting the optimal keys.
- To improve the privacy of data, an efficient chebyshev polynomial based ElGamal encryption (CPEE) is employed.
- To analyze the performance of proposed CPEE-CFGC algorithm by evaluating varied metrics and comparing the results with other existing methods for proving the efficacy.

The planning of the work is delivered as follows: Section 2 offers the literature review of the existing works, Section 3 expands on the proposed CPEE-CFGC algorithm, Section 4 discusses the results and analysis of the work and Section 5 accomplishes the conclusion and future scopes.

## 2. Related Works

Chunfu Zhang et al. [21] recommended an enhanced public key cryptography approach based on chebyshev polynomials and RSA. This algorithm addressed the drawbacks of earlier approaches by utilizing an alternate multiplication

coefficients and necessitating participants to disclose the specific value selection rules. Higher algorithm complexity was accomplished in the key generation and encryption/decryption phases by engaging more intricate intermediate process, which strengthened the method's defense against ordinaryattacks. Higher computation time, and lower efficacy in case of large prime bit sizes, were the shortcomings of this approach.

Sangjukta Das & Suyel Namasudra et al. [22] introduced an encryption strategy exploiting elliptic curve cryptography (ECC), advanced encryption standard (AES), and Serpentin order to safeguard healthcare data in IoT-enabled healthcare infrastructure. This hybrid encryption method, which had incorporated symmetric and asymmetric encryption algorithms, had enhanced the security measures of the medical data. Furthermore, an elliptic curve-based digital signature utilized in this method guaranteed the data integrity. Even with enhanced performance, the efficiency was low.

For secure data transfer, M. Indrasena Reddy et al. [23] offered enhanced ECC and chaotic mapping with fruitfly optimization (FOA). Todiminish the amount of input data, compression wasemgagedif the sender's plain text was first received. Complicated data was encrypted using the improved ECC (IECC) algorithm. The data was hidden by shuffling the pixels in the image through the usage of Chaotic mapping basedFOA, which supported the embedding of encrypted data. Consequently, there was diversity in the secret data. The limitations of this approach were higher encryption and decryption times.

Further, to progress the security of digital data, Mujeeb Ur Rehman [24] suggested a chaotic image encryption that was strengthened by means of quantum mechanics. By using quantum coding with 1-D sine-based chaotic maps (1-D SBCM) and adjusting the seed values, a random sequence was generated initially. Then, an enhanced quantum representation (EQR) model wasexploited to sensibly design a pseudorandom number generator (PRNG). Aquantum right cyclic shiftoperator and a quantum XOR operator were also utilized within this framework. These operators were essential to the formation of particularly durable encrypted images.

Moreover, to address the frequent difficulties with chaotic encryption techniques, M. Vijayakumar & A. Ahilan [25] presented a new encryption system based on chaotic map substitution boxes (S-box) and cellular automata (CA). This methodhad presented a 4D memristive hyperchaos with a more outstanding chaotic range, improved uncertainty, and ergodicity as an alternative to the software-based method, which was fragile and provided limited throughput, to resolve the insufficient randomness delivered by the 1D chaotic map. This couldmake this strategy less susceptible to manipulation.

### 2.1 Problem statement

In recent days, various methods have been introduced to address the requirement for strong encryption algorithms to defend sensitive data transferred through unsecure channels. Even if they are efficient, the conventional encryption methods mostly rely on the secrecy of shared keys to possess information private. Nonetheless, securely distributing and storing these keys becomes challenging, predominantly or when communicating parties are geographically distant or when frequent key changes are desirable. The risk to data secrecy modelled by symmetric encryption systems' exposure to key leakage is considerable. Public-key cryptography evolved as a solution to these issues and assisted symmetric encryption techniques overcome some of their drawbacks. RSA, ECC, Chebyshev polynomial based RSA etc., are some of the public-key cryptographic algorithm that delivers a more secure process of encryption by using key pairs, which are composed of a private key for decryption and public key for encryption. However, the secrecy and integrity of communication may still be threatened by potential susceptibilities like tampering and man-in-the-middle attacks, even with the improvements in public-key cryptography. Thereby, the goal of the research is to use Chebyshev polynomials and Elgammal algorithms to improve the dependability and security of public key encryption techniques. The proposed approach intents to reduce frequent attacks and progress the overall security of the cryptographic system by encrypting data using enhanced Chebyshev polynomials, which are distinctive in mathematics. This novel method pursues to deliver a more reliable algorithms for secure communication in different applications by addressing the limitations of existing public-key cryptography algorithms.

## 3. Proposed Methodology

This section combines the potential of Polynomial based ElGamal Encryption with Diffie- Hellman Key Exchange (CPEE-CFGC) for guaranteeing security in various applications. These techniques involve the key generation process followed by the stages of encryption and decryption and the secure key exchange process. The unique feature of proposed ElGamal encryption is the utilization of Chebyshev polynomials as a part of the encryption

process. Chebyshev polynomials are own family of orthogonal polynomials that possess potential things which make them appropriate for cryptographic algorithms. In addition to that Diffie-Hellman key exchange mechanism specifically used to securely create a shared secret key between two parties over an insecure channel primarily due to the discrete logarithm problem. During the initial step private keys are generated using Fuzzy Logistic Tent Membership Function (FLMF) for each party engaging in the communication, and the optimal keys are selected using Greater Cane Rat Algorithm (GCRA). In the process of shared secret key computation, the Diffie Hellman key exchange mechanism uses suitable parameters such as prime numbers and generators plus the modular exponentiation. In the encryption stage, the sender converts the plaintext into a polynomial; the choice of which polynomial to use is a random Chebyshev polynomial that helps increase security. All these components make up a 'ciphertext polynomial', which also contains a chosen random value. The sender then sends this ciphertext polynomial together with other parameters of encryption

After the ciphertext polynomial is received by the intended receiver, they decrypt the original plaintext message by applying the receiver's private key and the received parameters accordingly. This integration of cryptographic techniques improves the functionality of communication security by combining Diffie-Hellman's key exchange process with the encrypted strength of the Chebyshev Polynomial. If the opponent manages to intercept the communication and gain access to the public keys, it becomes extremely difficult for the opponent to decipher messages or extract the shared secret without the corresponding private keys, which makes the high level of security comparatively more achievable.The block diagram of the proposed method is given in Figure 1.

### 3.1 Private Key generation

In CPEE-CFGC algorithm, a unique key generation method, fuzzy logistic tent membership function (FLMF) that uses chaotic maps is engagedto generate keys for secure communication. The main intension of FLMFis to make intricate for the attackers to decode the encrypted information with no secret key understanding.The fuzzy logistic tent map, which is created by combining fuzzy triangular tent and logistic maps, produces a random series of numbers that can be employed as secure encryption keys. Here, a fuzzy triangle function is utilized to introduce an adaptation of the traditional tent map. The created key is used during the encryption process to improve the security and confidentiality of the data. As a result, by applying a variety of mathematical ideas, the space of potential secret keys is increased, building the system more resistant to attacks. The block diagram of FLMF for private key generation is provided in Figure 2.

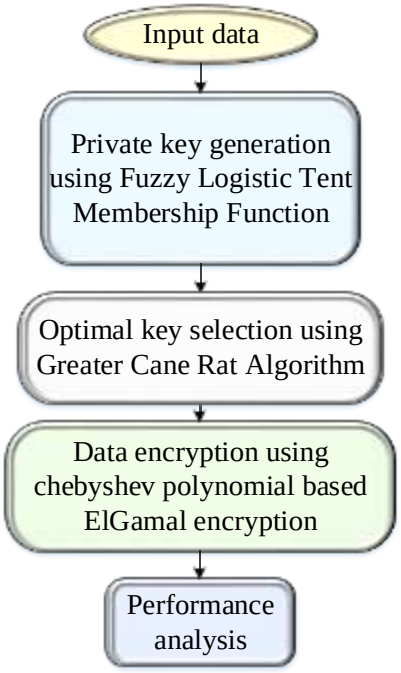


***Figure 1.*** *Block diagram of proposed CPEE-CFGCmethod*

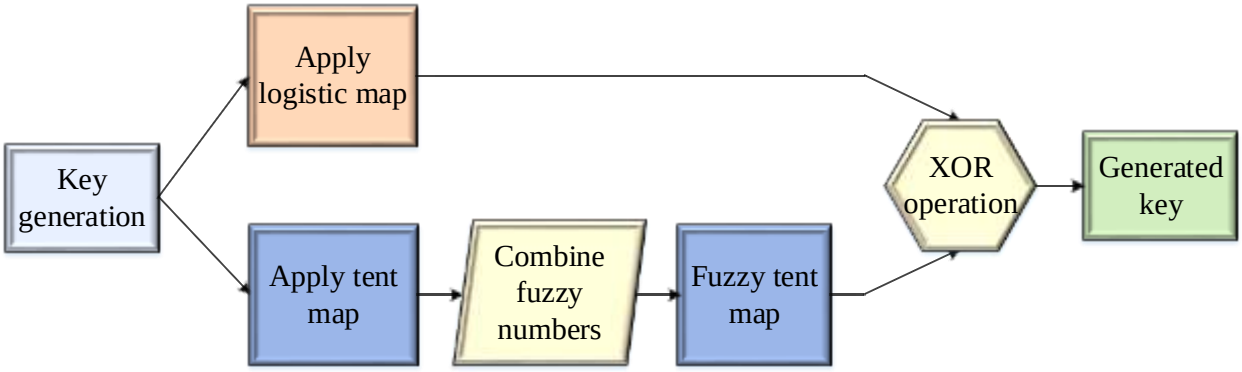


***Figure 2.*** *Block diagram of FLMFfor private key generation*

### 3.1.1 Logistic Map

A one-dimensional chaotic map with exceptional chaotic properties is termed as logistic map. The following describes the logistic map's numerical expression:

$$z_{(k)}(log\ j) = w_1 \times z_{(k-1)} \times (1 - z_{(k-1)}) \qquad (1)$$

where, $w_1$suggests the control parameter and its value range is [0,4]. The logistic map is exceptionally sensitive to initial condition $z_{(k)}$, which range from (0,1).

### 3.1.2 Tent map

The tent map shows chaotic behavior since it is a nonlinear dynamical system and is sensitive to parameter values and initial conditions. During encryption, the tent map can be utilized to generate pseudorandom number sequences that are engaged as encryption keys. The tent map's numerical equationis provided below:

$$a_{(j)} = \begin{cases} v_2 \times a_{(k-1)}, if 0 \le a_{(j-1)} < \frac{1}{2} \\ v_2 \times (1 - a_{(j-1)}), if \frac{1}{2} \le a_{(j-1)} \le 1 \end{cases} \qquad (2)$$

where, $v_2 \in (0.5,2)$ implies the parameter and its initial value range is (0,1).

### 3.1.3 Fuzzy number

Fuzzy numbers are one kind of mathematical model that is used to characterize imprecision and uncertainty in data. They are signified by a membership function that disperses a membership degree to each element in the discourse universe. The membership degree is a number that varies from0 to1, where 1 signifies full membership and0designates no membership. In the triangular fuzzy membership function, three parameters such as the lowest value $(c)$, middle value $(d)$ and maximum value $(e)$ that are located at the triangle peak are used to govern inputs. However, $c$ and $e$ are placed at the bottom of the triangle, by this means $c \le d \le e$. Now, the equation of triangular membership function is delivered as follows:

$$f(z,c,d,e) = Maxi\left(Mini\left(\frac{z-c}{d-c}, \frac{e-z}{e-d}\right), 0\right) \qquad (3)$$

By using the triangular membership function, the tent map and triangle membership values are joined to form the fuzzy tent map. This produces an arbitrary, complex sequence number that is utilized as a secret key during the encryption process. Furthermore, an addtional random sequence numbers that function as secret keys are created using a logistic map. Additionally, the parameters producedby means of the triangle membership function are used to modify the original keys map equations. This amalgamation can be used to generate optimal keys for data encryption.

### 3.1.4 Fuzzy tent map

Fuzzy mathematics concepts are extensivelyused to create extremely chaotic systems for encryptionprocessas tent maps can exhibit complex and unpredictable behavior. As a result, the CPEE-CFGC algorithm incorporates tent mapping with the fuzzy mathematics idea. During secure communication, the data is safely encrypted using the intricate behavior exhibited by the map. Using the advantages of both ideas, the CPEE-CFGC algorithm aims to provide more security. The following is the present implementation of fuzzy numbers to tent maps:

$$z_{(k+1)} = \begin{cases} w_2 \times f_{tri} \times b_{(k-1)}, if 0 \le b_{(k-1)} < \frac{1}{2} \\ w_2 \times f_{tri} \times (1 - b_{(k-1)}), if \frac{1}{2} \le b_{(k-1)} \le 1 \end{cases} \qquad (4)$$

where, $f_{tri}$suggests the fuzzy triangular.

### 3.1.5 Fuzzy logistic tent membership function

The secure keys for encryption is a sequence of random numbers created using a hybrid map and fuzzy concept. In order to enhance the security and confidentiality of data during the encryption process, the logistic map is reflected and utilized as a key. Fuzzy mathematics is used to provide values that affect initial keys in each iteration and produce new cipher text. To get the final key in FLMF, the keys created in the first phase using the logistic map and fuzzy tent map are combined using an XOR operation. Now, the final key generation expression is stated as follows:

$$E_{Key} = z_{(k)}(log\ j) \oplus z_{(k+1)} \qquad (5)$$

After generating the secret/private key, the optimal keys are selected and through the metaheuristic optimization algorithm.

### 3.2 Optimal key selection

After generating the private and public keys, the proposed CPEE-CFGC algorithm chooses the optimal keys using greater cane rat algorithm (GCRA).GCRA is one of the recent metaheuristic

approach employed for solving optimization issues. The clever foraging practices ofgreater cane rats (GCR) during and after mating season served as the inspiration for GCRA. GCRs are nocturnal and intelligent enough to leave trails in the grass and reeds where they forage. These trails would finally lead to sources of water, food, and shelter. The exploration stage will beginif theyleave from several shelters scattered around their area for foraging and leaving trails. It is presumed that the dominant male rodent keeps track of these routes, and other rats alter their position based on this information. Further, the males separate from the group if they identify the breeding season. The foraging activities are concentrated in areas with a surplus of food sources after the group separation, and facilitating exploitation. Consequently, to portray the design of GCRA and perform the optimization objectives, the incisive foraging paths and behaviors during the mating season are mathematically modeled. In addition, a comprehensive analysis of convergence and computational outcomes proved the effectiveness and stability of GCRA. Besides, the GCRA outperformed competing optimization algorithms by creating optimal or nearly optimal solutions and evading the trap of local minima. Due to these advantages, the proposed CPEE-CFGC algorithm has selected GCRA for optimal key selection. Like other optimization algorithms, GCRA generate the initial population of GCR, which means the keys are generated using the below equation.

$$Y = \begin{bmatrix} y_{1,1}y_{1,2} \cdots y_{1,d-1}y_{1,d} \\ y_{2,1}y_{2,2} \cdots y_{2,d-1}y_{2,d} \\ \vdots\vdots\ y_{j,k}\ \vdots\vdots \\ y_{p,1}y_{p,2} \cdots y_{p,d-1}y_{p,d} \end{bmatrix} \qquad (6)$$

where, $Y$ indicates the complete GCR population, $d$ and $p$ resemble the dimension of problem and the size of the population. Now, the below equation is utilized to randomly produce the individual rat $(y_{j,k})$ of$j^{th}$position in $k^{th}$dimension.

$$y_{j,k} = Rand(Ub_k - Lb_k) + Lb_k \qquad (7)$$

where, $Lb_k$ and $Ub_k$ designate the lower and upper bound in $k^{th}$ dimension, and$Rand$ represents the random number between 0 and 1.

A variable that controls whether or not it is a rainy season is specified as $\vartheta$. Depending on this $\vartheta$ value, the GCRA enters either the exploitation or the exploration stage. The $\vartheta$value is sensibly chosen to strike a balance between exploration and exploitation. After a thorough parametric analysis, the $\vartheta$ value is carefully adjusted to 0.5.

$$y_{j,k}^{New} = 0.7 * \frac{(y_{j,k}+y_{l,k})}{2} \qquad (8)$$

where,$y_{l,k}$represents the dominant male in $k^{th}$ dimension, $y_{j,k}$ designates the current position of GCR, and $y_{j,k}^{New}$indicates the new position of GCR.

### 3.2.1 Fitness computation

Forming a fitness function to measure each solution's performance is indispensable in order to estimate the fitness of solution. The below equation presents the formulation of fitness function utilized for optimal key selection.

$$FF = Maxi(kbt) \qquad (9)$$

where,$kbt$ indicates the key breaking time, $FF$ represents the objective function that yields the maximum key breaking timeas an optimal solution. Because of this, GCR solutions compute the key breaking time for each data matrix solution. If the solution accomplishes optimal fitness, it is terminated; if not, it uses the GCRA to update the public and private keys.

### 3.2.2 Exploration stage

The GCR construct their burrows/nest, which are shallow tunnels around their territory. When GCRs leave the different shelters to go foraging, they either scavenge for new food sources, or they follow trails to previous food sources and leave trails.The below equation shows how the dominating male's position defines a new position for the remaining rat population in the search area.

$$y_{j,k}^{New} = y_{j,k} + D \times (y_{l,k} - s \times y_{j,k}) \qquad (10)$$

where, $y_{l,k}$ represents the dominant male in $k^{th}$dimension,$D$ indicates the random number defined within the problem space boundaries, pretending the dispersed food sources and shelter.

During this stage of GCR motion simulation, the fittest rat is updated and the location of the other rats are altered in accordance with the newly compute fittest rat when the fitness of another rat exceeds the fittest rat. If not, it leaves from the position of fittest rat. This movement technique of GCR is expressed in the below equation.

$$Y_j = \begin{cases} y_{j,k} + D \times (y_{j,k} - \beta \times y_{l,k}), F_j^{New} < F_j \\ y_{j,k} + D \times (y_{n,k} - \chi \times y_{l,k}), Otherwise \end{cases} \qquad (11)$$

where, $Y_j$ represents the upcoming or new state of the $j^{th}$ GCR, $F_{y_l}$specifies the value of dominant

male's fitness function,$F_{y_j}$specifies the current value of the fitness function. $s$ mimics the influence of an abundant food source that tends to increased exploitation.

$$s = F_{y_l} - D_{iter} \times \left(\frac{F_{y_l}}{Maxi_{iter}}\right) \qquad (12)$$

A coefficient that mimics a dwindling food source and forces the search for new food or shelter is given as $\beta$ .

$$\beta = 2 \times s \times Rand - s \qquad (13)$$

A coefficient that forces the GCR to relocate to other available plentiful food sources within the breeding space is specified as $\chi$.

$$\chi = 2 \times s \times \varpi - s \qquad (14)$$

where, $D_{iter}$ resembles the current iteration, and $Maxi_{iter}$ represents the maximum iteration.

### 3.2.3 Exploitation stage

The breeding season generally happens during the wet season, and varies according on the habitat. During the breeding season, the males are well-known to scatter from the group. It is supposed that after group separation, the foraging activities will focus in regions with plentiful food sources. Thisstage is simulated initially by choosing a female $n$ at random such that $n \neq l$ (the dominant male). The intensification happens near the chosen female since breeding takes place near abundant foods sources. The modeling of the process is provided in below equation.

$$y_{j,k}^{New} = y_{j,k} + D \times (y_{l,k} - \varpi \times y_{n,k}) \qquad (15)$$

where, $y_{n,k}$ designates the location of randomly chosen female in $k^{th}$ dimension and $\varpi$randomly selects values between 1 and 4, imitating the number of offspring producedannually by each female GCR. The newly computed location for GCR takes precedence over the previous position if it maximizes the value of target function, as represented by equation (1).During the iterations, better exploration and exploitation are instigated by the parameters $D, s, \varpi, \vartheta, \beta$, and $\chi$.The pseudocode of GCRA for optimal key selection is provided in Algorithm 1.

**Algorithm 1:** Pseudocode of GCRA for optimal key selection
*Input:*Population of GCR, $\vartheta$, maximum iteration
*Output:* Optimal search outcome
***Start***
Initialize the size of population, and other parameters
Compute the fitness of each GCR
Choose the fittest GCR as the dominant male $Y_l$
Update the global best solution
Update the enduring GCR depending on the location of $Y_l$***For***$iter = 1: Maxi_{iter}$
Evaluate $D, s, \varpi, \beta, \chi$
***If***$Rand < \vartheta$
Exploration:
Update the position of current search agent
Check boundary constraints
***Else***
Exploitation:
Update the position of current search agent
Check boundary constraints
***End if***
Compute the fitness of each GCR depending on the new location
Update search agent
Update global best
Choose a new dominant male $Y_l$
***End for***
***Return*** global best solution
***End***

## 3.3 Data encryption and decryption

### 3.3.1 General public key cryptography algorithm

- Every user produces a key pair, $Ke = (Ke_d, Ke_e)$, in which $Ke_e$ signifies the public key and $Ke_d$ resembles the private key. Theoretically, $Ke_d$ must be calculated from$Ke_e$ for public key cryptography technique but in practice this is not possible because of greater computational complexity.
- The information sender utilized the information receiver's publicly avalikable key, $Ke_e$, to encrypt the plaintext as follows: $E(pl, Ke_e) = Ci$, where $pl$ indicates the plaintext and $Ci$ specifies the encrypted ciphertext.
- The information receiver uses their own privately held secret key, $Ke_d$, to decode the data: $Ke_d$: $D(Ci, Ke_d) = pl$.

### 3.3.2 Chebyshev polynomials with their properties

Let $Ci[-1,-1]$ be a vector space comprising of every continuous real-valued functions on$[-1,1]$, then $\{cos(p\,cos^{-1}(y))\}_{p=0}^{p=\infty}$ is a set of bases on $Ci[-1,1]$. Let

$$U_p = cos(p\,cos^{-1}(y)) \qquad (16)$$

then, $U_p$ is termed as $p$ order Chebyshev polynomial of first kind. It partakes the following properties:

- First,

$$U_0 = 1, U_1 = y, and\, U_{p+2}(y) = 2yU_{p+1}(y) - U_p(y) \qquad (17)$$

- Secondly, the mapping $\rho_p: y \to U_p$ on $[-1,1]$ is defined and then holds the following semi-group property.

$$\rho_{mp}(y) = \rho_m(\rho_p(y)) = \rho_p(\rho_m(y)). \qquad (18)$$

The properties are required to be extended to clearly state the design concept of chaotic key scheme.

*Theorem 1.* The semi-group property of Chebyshev polynomials is valid on the interval $(-\infty, +\infty)$;

*Theorem 2.* If $n$ is an odd prime number and $y$ is a positive integer, then

$$U_n(y) \equiv y(mod\, n); \qquad (19)$$

*Theorem 3:* Assume that $y$ and $e$ are positive integers and $n$ and $q$ are relatively prime. When $U_e(y) \equiv y(mod\, n)$ and $U_e(y) \equiv y(mod\, q)$, then $y$

$$U_e(y) \equiv y(mod\, n\, q) \qquad (20)$$

**Proof.** Since $(n, q) = 1$, there exist integers $t$ and $u$ such that $tn + uq = 1$, thereby, $U_e(y) = tnU_e(y) + uqU_e(y) \equiv tny + uqy \equiv y(mod\, n\, q)$.

**Corollary 1.** When $U_e(y) \equiv 1$ for every $y$, and there are positive integers $d$ and $e$ satisfying $ed \equiv 1(mod\, \phi)$, then

$$U_{ed}(y) \equiv y(mod\, n\, q) \qquad (21)$$

where, $n$ and $q$ are coprime primes.

### 3.3.3 Elgamal cryptosystem

Assume that $P$ and $Q$ are the two parties communicating with each other through an unsecured channel, with $P$ serving as the sender and $Q$ as the receiver.

*Key generation:* At first $Q$ defines the primitive root $g$ modulo $n$ by choosing a large prime number, $m$. ElGamal proposes that to ensure system security, $n$ should be selected so that $n-1$has at least one large prime factor. Let $y(1 \le y \le n-2)$ be the secret decryption key of $Q$ and $z(1 \le z \le n-2)$ be the secret encryption key of $P$. Now, $Q$ publishes $m, g,$ and $a$ after computing $a \equiv g^y(mod\, n)$.

Private keys: $y, z$

Public keys: $n, g, a$

*Encryption:*Now, the plain text $m$ encrypted by $P$ as follows.

$$b \equiv g^z(mod\, n) \qquad (22)$$

$$c \equiv m.a^z(mod\, n) \qquad (23)$$

Share $b$ and $c$ with $Q$.

*Decryption:* After receiving $b$ and $c$, $Q$ recovers $m$.

$$b^y \equiv a^z(mod\, n) \qquad (24)$$

$$m \equiv c.(\overline{a^z})(mod\, n) \qquad (25)$$

where, $(\overline{a^z})$ represents the inverse of $a^z$ under modulo $n$.

### 3.3.4 Chebyshev Polynomial based ElGamal Encryption

The diffie-hellman key exchange (DHKE) is the basis of public-key cryptosystem well-known as ElGamal encryption algorithm. Discrete logarithms and modular exponentiation are utilized for both encryption and decryption processes. The chebyshev polynomial based ElGamal encryption (CPEE) can offer improved security features in combination with Chebyshev polynomials and other multiplication factors. The ElGamal encryption scheme using Chebyshev polynomials and alternative multiplication factors is described below as follows:

- ***Key generation:***Using generator $g$, a cyclic group $G$ of order $n$ is created. It is practicable to create every group element in a cyclic group with the powers of one of its own elements. Next, $h = g^y$ is determined for randomly selected $y \in Z_n^*$. Eventually, the public key is $(G, n, g, h)$ and secret key is resembled as$x$. Here, the public key is determined through the Chebyshev polynomial.
- ***Encryption:***The message $M$is encrypted by utilizing $g$ and $x$, which is selected through GCRA. The

encryption procedure outputs a pair of ciphertexts, $(c = (c_1, c_2))$:

$$c = E(M) = (g^x, Mh^x) = (g^x, Mg^{xy}) = (c_1, c_2) \quad (26)$$

- ***Decryption:*** First, compute $s = {c_1}^y$, where $y$ represents the secret key, is computed in order to decipher the ciphertext $c$. The decoding algorithm then functions as follows:

$$c_2.s^{-1} = Mg^{xy}.g^{-xy} = M \quad (27)$$

### 3.4 Diffie-hellman key exchange

The DHKE algorithm is a key exchange algorithm that is used in the proposed CPEE-CFGC algorithm to securely exchange the encrypted keys. The DHKE techniqueis an important constituent of modern cryptography. Itpermits two parties to securely exchange keys for encryption across an unstable communication channel. A huge prime integer and a primitive root are the two public parameters for the DHKE approach that both parties agree upon. Each party produces a private key for the both parties. Each party states its public key conforming to these specifications by raising the primitive root to the power of its private key modulo. These private keys are kept confidential. The publicly generated keys are exchanged by the parties and are produced significantly. Each party individuallycalculates the shared secret key by expanding the attained public key to modulo of its own private key after obtaining the public key of the other party.This shared secret key serves as the encryption key for their communication and is obtained by merging the public key of one party with the private key of other. Furthermore, an attacker cannot compute obtain the shared secret key from the exchanged public keys because of the discrete logarithm problem. In the event that communication is intercepted, this offers DHKE security. Accordingly, parties can exchange keys securely while maintaining the confidentiality and integrity of their communications using DHKE. The structure of DHKE algorithm for key exchange is provided in Figure 3. The following provides the steps that involved in DHKE algorithm for exchanging keys.

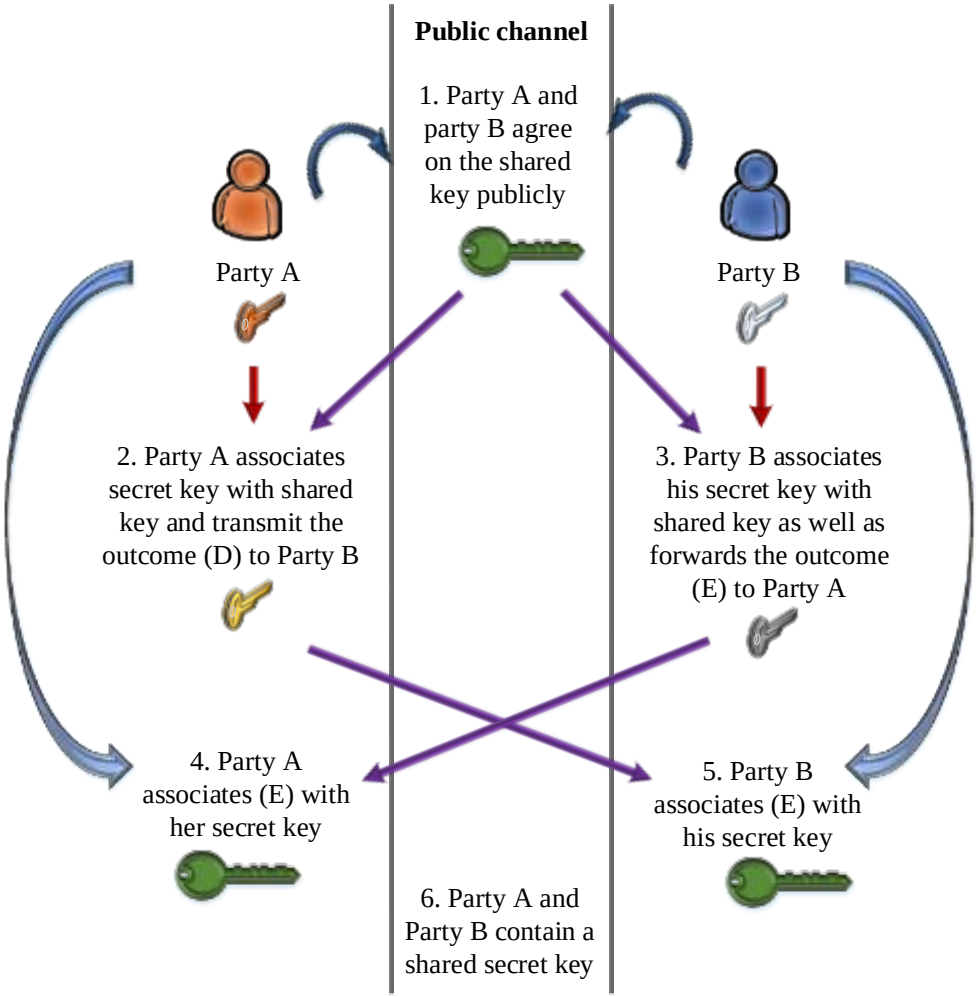


***Figure 3.*** *DHKE algorithm for key exchange*

*Parameter setup:* Two public parameters, specifically a large prime integer $q$ and a primitive root $h$ modulo $q$, are agreed by both parties.These parameters can knowby both partieseven if they are not secret. Moreover, each party produces their own set of private keys. These keys are symbolized to $b$ for party $P$, and $c$for party $Q$. These private keys are not exposed to any parties and are kept confidential.

*Computation of public key:*Party $P$ computes its public keyby exploiting the primitive root $h$ to the power of its private key a modulo $q$. In mathematical expressions, it is stated as $P = h^b \, mod \, q$. Likewise, to computed its public key, party $Q$exploits the primitive root $h$upraised to the power of its private key $C$ modulo $q$. Statistically, it is quantified as $Q = h^c \, mod \, q$. Next, the public keys $P$ and $Q$ are exchanged by both parties through the unsecure communication channel.

*Calculation of shared secrets:* Following the party $Q's$ public key distribution to Party $P$, it computes the shared secret key, $S$ by maximizing$Q$ to the power of its private key, $b$modulo $q$. It is scientificallyuttered as: $S = Q^b \, mod \, q$. Correspondingly, upon finding party $P's$public key, $P$, party $Q$analyses the shared secret key, $S$by exploiting$P$ to the power of its private key, $c$ modulo $q$. In mathematical viewpoint, it is quantified as $S = P^c \, mod \, q$. Now, both the parties have the identical secret key, $S$, which is used for asymmetric encryption as well as decryption.

*Key Exchange:*Both the party $P$ and party $Q$contain the shared secret key. The shared secret key is the identical for both parties for the reason that it is independentlycreated by each party based on their private and the public key of the other party. Parties $P$ and $Q$ can remain communicating using this shared secret key as the assymmetric encryption key. Accordingly, in proposed method, DHKE delivers a secure method for exchanging keys over an unstable communication channel between two parties.

## 4. Results and Discussion

The proposed CPEE-CFGC performed the security examination on the enhanced public key encryption algorithm. Rendering to the theoretical evaluation, the CPEE-CFGC algorithm is a public key cryptosystem based on Elgammal algorithm with optimal key selection and chebyshev polynomials that deliberates both of their security aids and is capable of withstanding common attacks. Moreover, the attackers cannot exploit the Chebyshev polynomials' periodicity to break it since the cosine representation of Chebyshev polynomials specified on the interval $(-\infty, +\infty)$is invalid. It can also endure popular modular attacks. The linear independence between plaintexts cannot be preserved after encryption transformation and this makesCPEE-CFGC resistant to low exponent attacks. The execution of proposed CPEE-CFGC algorithm is made using the python programming language. The analysis and implementation of CPEE-CFGC algorithm are carried out on Intel® core™ i5-10210U CPU @ 1.60GHz 2.11 GHz 8 GB (7.79 GB usable), with windows 10 operating system.

### 4.1 Performance Evaluation

In this section, the outcome of the proposed CPEE-CFGC algorithm is tested and compared with existing methods to determine the superiority of providing securing in data communication. Different evaluation metrics such as key generation time, encryption time, decryption time and total execution time are used in CPEE-CFGC algorithm and they are examined on dissimilar bit sizes of initial primes, varying from 64 to 2048. Key generation time is considered as a significant metrics that influence the practicability and efficiency of CPEE-CFGC algorithm. Figure 4 describes the comparative analysis of key generation time with proposed CPEE-CFGC and other existing algorithms for varying bit sizes of 64, 128, 256, 512, 1024 and 2048.

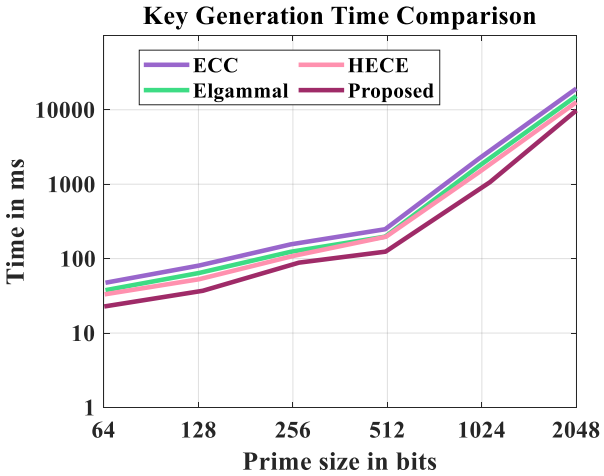


***Figure 4.*** *Analysis of key generation time*

In contrast to existing techniques, the proposed CPEE-CFGC algorithm shows better enhancement in key generation time for varying number of bit sizes. The outcomes of this comparative analysis illustrate that the efficiency of CPEE-CFGC algorithm maximizes with bit size. Among the existing algorithms like elliptic curve cryptography (ECC), Elgammal, and hybrid elliptic curve elgammal (HECE),HECE work reasonably well for

varying bit sizes, with key generation time falling within reasonable bounds. However, the key generation time for this existing technique significantly increases if the bit size reaches to 2048 bits and beyond, often ensuing in delays and lessened performance in the perspective of secure communications. Conversely, the GCRA in proposed CPEE-CFGC algorithm has produced keys more quickly even with maximizes bit sizes because it uses the sophisticated mathematical formulas and efficient computation methods.

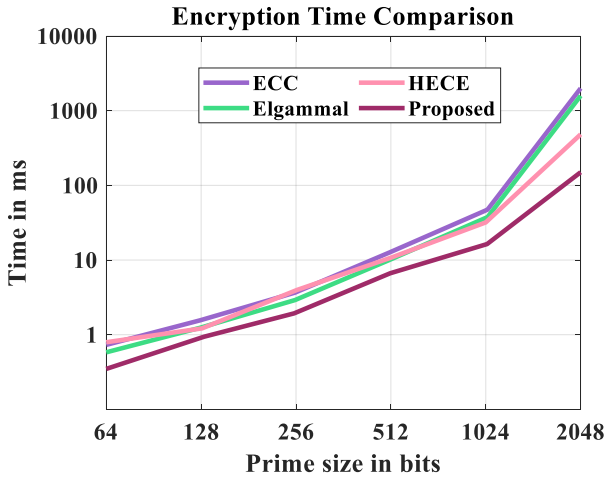


***Figure 5.*** *Analysis of encryption time*

Another important criterion for evaluating the efficacy of proposed CPEE-CFGC algorithm is encryption time. The encryption time has a direct impact on models responsiveness and data processing efficiency in the perspective of secure communication. Figure 5 compares the encryption times with CPEE-CFGC and existing algorithms by changing the bit size. In the graphical representation, it is noticed that the encryption time of CPEE-CFGC is lower for varying bit sizes. By utilizing FLMF and enhanced chebyshev polynomial-based Elgammal's computational efficiency, the proposed CPEE-CFGC algorithm upheld robust security measures while decreasing encryption time compared to existing approaches such as ECC, HECE, and Elgammal. The CPEE-CFGC algorithm effectively created cryptographic keys, which are essential for guarding data by applying FLMF with GCRA for key generation. FLMF with GCRA generates keys speedily and with little computer overhead through the usage of relatively simple mathematical processes. Accordingly, adding chebyshev polynomial-based Elgammal for encryption further increases the efficiency. Moreover, this allows for faster encryption times without losing security. Overall, CPEE-CFGC algorithmgreatly declines encryption time by optimizing keys and encryption with chebyshev polynomial-based ElGamal.

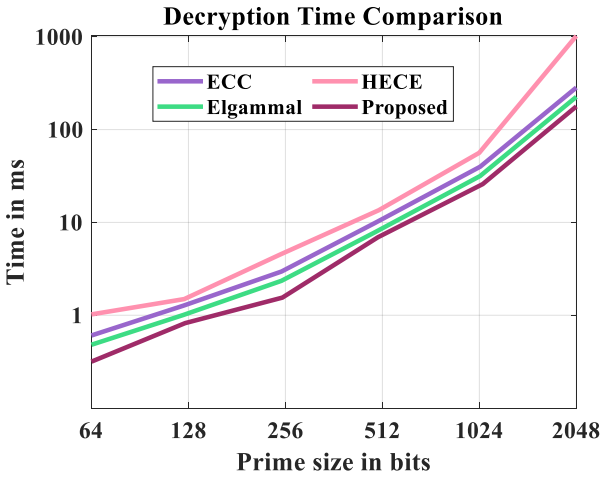


***Figure 6.*** *Analysis of decryption time*

Like encryption time, the decryption time is also a fundamental performance indicator. The efficacy and response time of the proposed CPEE-CFGC algorithm are directly impacted by the decryption time. The strength of CPEE-CFGC algorithm for data encryption is accomplished by incorporating ElGamal and chebyshev polynomial to provide strong security with the least amount of computational overhead. The success of chosen CPEE-CFGC algorithm is crucial in terms of decryption time. Figure 6 designates the comparison of decryption time with proposed CPEE-CFGC and exiting algorithm by varying the prime size in bits. In contrast to the prevailing encryption methods, the proposed CPEE-CFGC algorithm allows for faster decryption owing to its better key selection and computing efficiency.

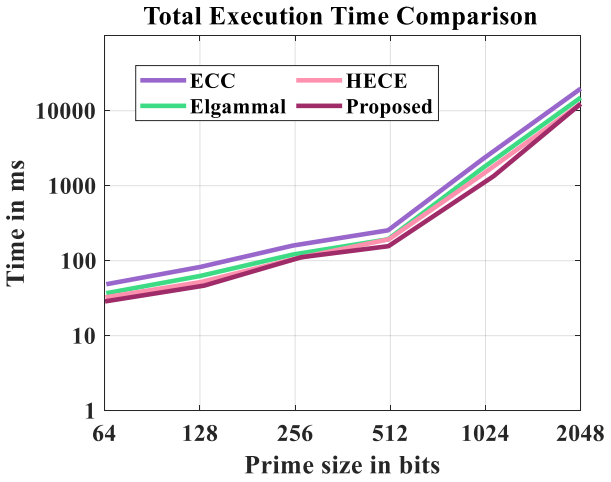


***Figure 7.*** *Analysis of total execution time*

Figure 7 specifies the performance comparison of total execution time by varying the bit sizes. In the graphical demonstration, it is witnessed that when the size of bits maximizes, the execution time also upsurges. Nevertheless, the total execution time taken to 1024 and 2048 are significantly less. The proposed CPEE-CFGC algorithmhas utilized an

execution time of 35.48 ms, 55.24 ms, 110.56 ms, 195.45 ms, 1300.56 ms and 12100.56 ms, while processing 64 bits, 128 bits, 256 bits, 512 bits, 1024 bits and 2048 bits. Moreover, this analysis demonstrates that the proposed CPEE-CFGC algorithm consistently outperforms conventional algorithms. Table 1 designates the comparative outcomes of the proposed CPEE-CFGC algorithm and existing methods by changing the bit sizes.

***Table 1.*** *Comparative analysis of CPEE-CFGC and existing methods by changing the bit sizes*

| **Methods** | **Key generation time (ms)** | | | | | |
|---|---|---|---|---|---|---|
| | **64 bits** | **128 bits** | **256 bits** | **512 bits** | **1024 bits** | **2048 bits** |
| ECC | 70.12 | 96.41 | 200.31 | 400.54 | 3600.35 | 30010.25 |
| Elgammal | 60.14 | 81.21 | 150.47 | 290.54 | 3003.24 | 18005.79 |
| HECE | 65.12 | 73.4 | 120.65 | 270.98 | 2000.25 | 15890.12 |
| **Proposed** | **35.14** | **53.14** | **92.69** | **130.45** | **980.56** | **10256.25** |
| **Methods** | **Encryption time (ms)** | | | | | |
| | **64 bits** | **128 bits** | **256 bits** | **512 bits** | **1024 bits** | **2048 bits** |
| ECC | 0.93 | 2.54 | 5.86 | 15.56 | 65.49 | 3000.54 |
| Elgammal | 0.78 | 1.54 | 4.95 | 10.32 | 58.71 | 2000.56 |
| HECE | 0.95 | 1.45 | 6.65 | 12.45 | 50.45 | 680.47 |
| **Proposed** | **0.55** | **0.98** | **2.46** | **8.347** | **20.87** | **160.78** |
| **Methods** | **Decryption time (ms)** | | | | | |
| | **64 bits** | **128 bits** | **256 bits** | **512 bits** | **1024 bits** | **2048 bits** |
| ECC | 0.78 | 1.65 | 4.93 | 12.45 | 60.65 | 450.89 |
| Elgammal | 0.7 | 1.56 | 6.82 | 9.24 | 50.34 | 560.11 |
| HECE | 1.26 | 2.65 | 4.65 | 18.32 | 75.69 | 1350.55 |
| **Proposed** | **0.52** | **0.93** | **2.13** | **8.79** | **38.75** | **230.45** |
| **Methods** | **Total execution time (ms)** | | | | | |
| | **64 bits** | **128 bits** | **256 bits** | **512 bits** | **1024 bits** | **2048 bits** |
| ECC | 69.12 | 92.45 | 203.45 | 400.8 | 3856.23 | 25000.25 |
| Elgammal | 55.35 | 86.45 | 140.56 | 310.54 | 2501.56 | 18324.55 |
| HECE | 52.65 | 70.58 | 128.65 | 300.45 | 1900.45 | 12530.57 |
| **Proposed** | **35.48** | **55.24** | **110.56** | **195.45** | **1300.56** | **12100.57** |

***Table 2.*** *Comparative analysis of CPEE-CFGC and other state-of-the-art methods by changing the bit sizes*

| **Model** | **Length of primes (in bits)** | **Key generation time (in ms)** | **Encryption time (in ms)** | **Decryption time (in ms)** | **Total execution time (in ms)** |
|---|---|---|---|---|---|
| RSA | 64 | 20.27 | 0.26 | 0.22 | 20.75 |
| | 128 | 25.47 | 0.37 | 0.33 | 26.17 |
| | 256 | 40.5 | 0.97 | 0.85 | 42.32 |
| | 512 | 76.55 | 1.75 | 1.68 | 79.99 |
| | 1024 | 820.14 | 15.28 | 14.54 | 849.96 |
| | 2048 | 4575.03 | 52.26 | 37.39 | 4664.67 |
| XRSA | 64 | 32 | 0.57 | 0.54 | 33.11 |
| | 128 | 47.61 | 1.3 | 1.05 | 49.95 |
| | 256 | 93.54 | 2.02 | 2.01 | 97.56 |
| | 512 | 188.91 | 10.53 | 9.56 | 209.00 |
| | 1024 | 922.81 | 39.68 | 36.06 | 998.56 |
| | 2048 | 8706.22 | 185.98 | 221.41 | 9113.61 |
| CRPKC-Ki | 64 | 35.02 | 0.65 | 0.64 | 36.32 |
| | 128 | 54.19 | 1.29 | 1.08 | 56.57 |
| | 256 | 105.99 | 3.12 | 3.09 | 112.19 |
| | 512 | 177.18 | 10.93 | 10.01 | 198.12 |
| | 1024 | 1250.09 | 39.87 | 39.24 | 1329.19 |
| | 2048 | 10,036.74 | 1896.98 | 235.31 | 12,169.03 |
| **Proposed** | **64** | **35.14** | **0.55** | **0.52** | **35.48** |
| | **128** | **53.14** | **0.98** | **0.93** | **55.24** |
| | **256** | **92.69** | **2.46** | **2.13** | **110.56** |
| | **512** | **130.45** | **8.347** | **8.79** | **195.45** |
| | **1024** | **980.56** | **20.87** | **38.75** | **1300.56** |
| | **2048** | **10256.25** | **160.78** | **230.45** | **12100.57** |

### 4.2 Comparison with other state-of-the-art methods

In this section, the results of the proposed CPEE-CFGC algorithm is compared with other state-of-the-art methods such as effective and enhance RSA (XRSA) [1], modified and secure RSA-based model (MRSA) [2],enhanced and secured RSA key generation scheme (ESRKGS) and Chebyshev-RSA public key cryptography with multiplication factor (CRPKC-Ki). In [1], the RSA method has enhanced in order to generate a more complex key pair such as a public and private key such that an adversary could never be capable to identify the private key using the public key. Here, the public and private key pairs are generated using four randomly chosen large prime integers. During the key-generation, encryption and decryption stages, this method also employed XOR operation in conjunction with a more intricate intermediate step to maximize algorithm complexity. MRSA algorithm based on a separate prime number of "n" has presented in [2]. Here, the algorithm's complexity has maximized due to the presence of prime integer "n" that make it harder to factor the variable "N". By utilizing a double encryption-decryption process, MRSA generated two distinct public keys and a private key from the huge factor of the variable "N," offering maximized security. Similarly, in [3], an ESRKGS that depend on a separate prime number of "n" hasintroduced. Here, the decryption and encryption keys are based on the product of four large prime numbers and the public key component is considered as the product of two large prime numbers to maximize the security. In [4], CRPKC has offered by employing an alternative multiplication coefficient to forge ciphertext. Moreover, this method has assisted to resist common attacks. Table 2 designates the comparative outcomes of the proposed CPEE-CFGC algorithm and other state-of-the-art methods by changing the bit sizes.

## 5. Conclusion

This papers contributes a novel CPEE-CFGC for guaranteeing security in various applications. Key generation, encryption and decryption with secure key exchange process are the various steps involve in CPEE-CFGC algorithm. During key generation, the private keys are generated using Fuzzy Logistic Tent Membership Function (FLMF) for each party engaging in the communication. Then, to select the optimal keys Greater Cane Rat Algorithm (GCRA) has utilized. The Diffie Hellman key exchange mechanism is exchange the keys in an unsecure channel. Further, chebyshev polynomial based ElGamal encryption (CPEE) is used for the process of encryption. The simulation of CPEE-CFGC algorithm is done through python programming language, and the performance is evaluated with significant performance indicators. Accordingly, the CPEE-CFGC has attained a better key generation time of 10256.25 ms, encryption time of 5160.78 ms, decryption time of 230.45 ms and total execution time of 12100.57ms by varying the bit size to 2048 bits than the existing algorithms. In future, the proposed method will be extended by utilizing hybrid encryption strategies with enhanced optimization strategy. Besides, considering into the addition of extra security measures, such as multi-factor authentication, can support the system's protection against possible attacks.

## Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

## References

[1] Khan M., Alanazi A. S., Khan L. S., & Hussain I. (2021). An efficient image encryption scheme based on fractal Tromino and Chebyshev polynomial. *Complex & Intelligent Systems*. 7(5);2751-2764. https://doi.org/10.1007/s40747-021-00460-4

[2] Thushara G. A. & Mary Saira Bhanu S. (2023). Chebyshev Chaotic Map with Attribute Based Encryption on Session Based Data Sharing in Fog Environment. *Peer-to-Peer Networking and Applications.* 18(1). https://doi.org/10.1007/s12083-024-01841-5

[3] Sun J., Zhang P., & Kong X. (2023). Identity Authentication Protocol of Smart Home IoT based on Chebyshev Chaotic Mapping. *International Journal of Advanced Computer Science and*

*Applications*. 14(4). https://doi.org/10.14569/ijacsa.2023.0140461

[4] Yupapin P., Meshram C., Barve S. K., Ibrahim R. W., & Akbar M. A. (2023). An efficient provably secure verifier-based authentication protocol using fractional chaotic maps in telecare medicine information systems. *Soft Computing*. 27(10);6033-6047. https://doi.org/10.1007/s00500-023-07889-4

[5] Jiang M. & Yang H. (2023). Image Encryption Algorithm Using Multi-Level Permutation and Improved Logistic–Chebyshev Coupled Map. *Information*. 14(8);456. https://doi.org/10.3390/info14080456

[6] Singh S., Sharma P. K., Moon S. Y., & Park J. H. (2024). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*. 1-18. https://doi.org/10.1007/s12652-017-0494-4

[7] Adeniyi A. E., Jimoh R. G., & Awotunde J. B. (2024). A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security. *Computers and Electrical Engineering*. 118;109330. https://doi.org/10.2139/ssrn.4683742

[8] Manikandaprabhu P. & Samreetha M. (2024). A Review of Encryption and Decryption of Text Using the AES Algorithm. *International Journal of Scientific Research and Engineering Trends*. 10(2);400-404. https://ijsret.com/wp-content/uploads/2024/03/IJSRET_V10_issue2_195.pdf

[9] Adeniyi E. A., Falola P. B., Maashi M. S., Aljebreen M., & Bharany S. (2022). Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions. *Information*. 13(10);442. https://doi.org/10.3390/info13100442

[10] El-Douh A. A., Lu S. F., Elkony A., & Amein A. S. (2022). A systematic literature review: The taxonomy of hybrid cryptography models. *Advances in Information and Communication Conference*. 714-721. https://doi.org/10.1007/978-3-030-98015-3_49

[11] Rawat A. & Tiwari A. A Review on Cryptography. (2023). *Conference: International Conference on Recent Trends in Engineering & Technology (ICRTET 2023).* https://www.researchgate.net/publication/381828530_A_Review_on_Cryptography

[12] Somsuk K. & Sanemueang C. (2021). Increasing security to public key cryptography for point-to-point communication. *Journal of Discrete Mathematical Sciences and Cryptography*. 1-15. https://doi.org/10.1080/09720529.2021.1930656

[13] Zhang B. & Liu L. (2023). Chaos-based image encryption: Review, application, and challenges. *Mathematics*. 11(11);2585. https://doi.org/10.3390/math11112585

[14] Meshram A., Wazalwar N. M., & Meshram C. (2024). New Secure Password-based Authentication Procedure using Chebyshev Chaotic Maps. *IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*. 2;1-6. https://doi.org/10.1109/iatmsi60426.2024.10503268

[15] Muhammad A. S. & Özkaynak F. (2021). SIEA: secure image encryption algorithm based on chaotic systems optimization algorithms and PUFs. *Symmetry*. 13(5);824. https://doi.org/10.3390/sym13050824

[16] Lu Q., Yu L., & Zhu C. (2022). Symmetric image encryption algorithm based on a new product trigonometric chaotic map. *Symmetry*. 14(2);373. https://doi.org/10.3390/sym14020373

[17] Dai W., Xu X., Song X., & Li G. (2021). Audio encryption algorithm based on chen memristor chaotic system. *Symmetry*. 14(1);17. https://doi.org/10.3390/sym14010017

[18] Alsaif H., Guesmi R., Kalghoum A., Alshammari B. M., & Guesmi T. (2023). A novel strong S-box design using quantum crossover and chaotic boolean functions for symmetric cryptosystems. *Symmetry*. 15(4);833. https://doi.org/10.3390/sym15040833

[19] Shahna K. U. (2023). Novel chaos based cryptosystem using four-dimensional hyper chaotic map with efficient permutation and substitution techniques. *Chaos, Solitons & Fractals*. 170;113383. https://doi.org/10.1016/j.chaos.2023.113383

[20] Rahul B., Kuppusamy K., & Senthilrajan A. (2023). Dynamic DNA cryptography-based image encryption scheme using multiple chaotic maps and SHA-256 hash function. *Optik*. 289;171253. https://doi.org/10.1016/j.ijleo.2023.171253

[21] Zhang C., Liang Y., Tavares A., Wang L., Gomes T., & Pinto S. (2024). An Improved Public Key Cryptographic Algorithm Based on Chebyshev Polynomials and RSA. *Symmetry*. 16(3);263. https://doi.org/10.3390/sym16030263

[22] Das S. & Namasudra S. (2022). A novel hybrid encryption method to secure healthcare data in IoT-enabled healthcare infrastructure. *Computers and Electrical Engineering*. 101;107991. https://doi.org/10.1016/j.compeleceng.2022.107991

[23] Reddy M. I., Reddy M. P., Reddy R. O., & Praveen A. (2024). Improved elliptical curve cryptography and chaotic mapping with fruitfly optimization algorithm for secure data transmission. *Wireless Networks*. 30(3);1151-1164. https://doi.org/10.1007/s11276-023-03554-8

[24] Rehman M. U. (2024). Quantum-enhanced chaotic image encryption: Strengthening digital data security with 1-D sine-based chaotic maps and quantum coding. *Journal of King Saud University-Computer and Information Sciences*. 36(3);101980. https://doi.org/10.1016/j.jksuci.2024.101980

[25] Vijayakumar M. & Ahilan A. (2024). An optimized chaotic S-box for real-time image encryption scheme based on 4-dimensional memristive hyperchaotic map. *Ain Shams Engineering Journal*. 15(4);102620. https://doi.org/10.1016/j.asej.2023.102620

[26] Imam R., Anwer F., & Nadeem M. (2022). An effective and enhanced RSA based public key encryption scheme (XRSA). *International Journal*

*of Information Technology*. 14(5);2645-2656. https://doi.org/10.1007/s41870-022-00993-y

[27] Islam M. A., Islam M. A., Islam N., & Shabnam B. (2018). A modified and secured RSA public key cryptosystem based on "n" prime numbers. *Journal of Computer and Communications*. 6(03);78. https://doi.org/10.4236/jcc.2018.63006

[28] Thangavel M., Varalakshmi P., Murrali M., & Nithya K. (2015). An enhanced and secured RSA key generation scheme (ESRKGS). *Journal of information security and applications*. 20;3-10. https://doi.org/10.1016/j.jisa.2014.10.004

[29] Zhang C., Liang Y., Tavares A., Wang L., Gomes T., & Pinto S. (2024). An Improved Public Key Cryptographic Algorithm Based on Chebyshev Polynomials and RSA. *Symmetry*. 16(3);263. https://doi.org/10.3390/sym16030263